PSA

SOFTWARE SYSTEM

Pixel - oriented System for Layout Analysis

USER MANUAL

J. Bloedel, R. Hartenstein, R. Hauck
M. Ryba, H.Salzmann, M. Weber

Departement of Computer Science
Kaiserslautern University
D-6750 Kaiserslautern

CONTENTS

CHAPTER 1

INTRODUCTION

PISA is a pixel-oriented system for layout and image analysis. The current version runs under VAX / VMS 3.7. The System is implemented in standard PASCAL (ISO level 0). The adaptation of the PISA SYSTEM to other operating system is an easy task.

The input structure is Manhatten CIF 2.0 with some useful extensions (orthogonal wires and polygons, user extension 94 and Berkeley user extension 9).

Currently three design technologies are supported :

- NMOS Mead & Conway Design Rules (buried layer included, without butting contact, 2.5 my)

- NMOS Design Rules of Dortmund University (without butting contact, 1.5 my)

- CMOS Design Rules of Dortmund University (without butting contact, 1.5 my)

An expansion of the PISA SYSTEM by other design technologies is possible.

The user interface is a menuedriven hierarchical dialog. It only presumes some basic commands for filehandling to be known by the user. A detailed description of the menue assembly can be found in the following chapters.

The next chapter demonstrates how to work with the PISA SYSTEM.

CHAPTER 2

EXAMPLE SESSION

2.1 BATCH-ORIENTED DESIGN RULE CHECK

To reproduce the session described, copy the file "regarray.cif" available on directory disk$pisa: to your current working directory. That's why the VMS command reads as follows :

```
$ copy disk$pisa:regarray.cif *.*
```

Start the PISA SYSTEM using the VMS command :

```
$ run disk$pisa:pisa
```

PISA requests

```
Is your terminal a VT101 or compatible (Y/N) ? _
```

Type "Y<ret>" if Your terminal is a DEC-VT101 or compatible, otherwise type "N<ret>". During the following session a VT101 compatible terminal is presumed.

PISA welcomes You with a screen shown in figure 2.1. Pressing return reveals the mask of Top Command Level (see figure 2.2). The first command (capitals distinguish legal abbreviations) should be :

```
DEfault set<ret>
```

A complete filename under VMS consists of device, directory, filename and extension. These are required by the DEfault set command of the PISA SYSTEM. We recommend urgently the specification of both directory and filename. The extension is generated automatically.

Enter for this session :

```
device    : <ret>
directory : <working directory>
filename  : regarray
```

See figure 2.3.

Following the last input Top Command Level is redisplayed.

A CIF description of cell "regarray" is available in file regarray.cif. This cell is to be proved for design rule violations. The command Autocheck performs a batchoriented design rule check.

    Autocheck<ret>

Two input lines are required :

- The filename of the cell to be checked. Enter <ret> to keep the default values set by the DEfault set command (see figure 2.4).

- The technology of the cell design. For this session MEAD/CONWAY is necessary. Enter :

    1<ret>

  (see figure 2.5)

Using these informations a batch job is generated. The VMS message

    JOB XX ENTERED ON QUEUE SYS$BATCH

occurs on the terminal (see figure 2.6). The user interactions are now completed. The bell announces the completion of the batch job. The following message appears on the screen.

    BATCH JOB XX regarray COMPLETED ON <date>

The following files were generated :

- regarray.lis listing of Design-Rule-Check

- regarray.plg logfile of PISA

If design rules were violated the additional files are generated :

- regarray.ecf errormarks in CIF format

- regarray.efm errormarks in FEM format

- regarray.plt plotfile with errormarks

Leave PISA SYSTEM by typing

    ENd<ret>

Statistics about cputime and elapsed time are displayed (see figure 2.7).

2.2 ERRORDISPLAY USING THE KIC EDITOR

The cif formatted error file (regarray.cif) supports the visual errordetection and errorcorrection using the layout editor KIC. The KIC editor only accepts kic formatted input. Therefor use the CIFtoKIC translator. CIFtoKIC requires the following input :

- Specification of the program the CIF file was generated by. Enter "KIC".

- Microns per lambda (here : 2.5).

- Input file (here : regarray.ecf)

For distinction of error symbols from layout symbols the kic-formatted error file is suffixed by "err" (here : regarrerr.;1).

Errordisplay using the KIC editor needs following inputs :

- EDit regarrerr<ret>

  Arrows pointing to incorrect layoutsections appear on the screen.

- Insta
  MASter regarray<ret>
  <ctrl>e 0 0<ret>

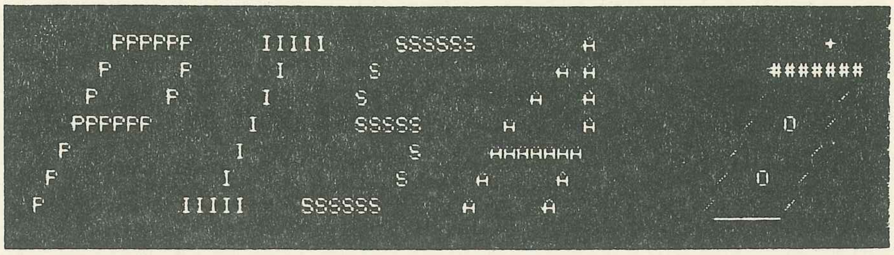
  Now the layout is specified as a subcell of the errorcell (regarrerr). Using the KIC command "Peek" the design rule violations are marked.

The KIC commands "SElect", "PUsh" and "POP" allow the correction of the layout cell.

Figure 2.8 shows a stipple plot following the KIC display.

2.3 ILLUSTRATIONS

Welcome to



(C) Copyright 1985 by
J. Bloedel, R. Hauck, M. Ryba
H. Salzmann, M. Weber

AG Hartenstein
Kaiserslautern University

Press <ret> to continue

figure 2.1 : PISA Welcome

```
                    PISA SOFTWARE SYSTEM 4.1
   (C) Copyright 1985 by AG Hartenstein Kaiserslautern University

                        TOP COMMAND LEVEL

 A (utocheck)      :  process specified CIF-source automatically
 C (olourplot)     :  generate plotfile from given CIF-file
 M (atrixplot)     :  generate stipfile from given CIF-file
 EX (perimental)   :  interactiv processing
 ED (it)           :  invoke VAX/VMS EDIT/EDT Editor
 DE (fault-set)    :  set filename default values
 SY (stem)         :  invoke VMS
 H (elp)           :  display further information
 EN (d)            :  exit

 Command           :  de

        started      at:    15:07:11.53          on:    19-MAR-1985
 device     :
 directory  :
 filename   :
```

figure 2.2 : PISA Top Command Level

```
                    PISA SOFTWARE SYSTEM 4.1
   (C) Copyright 1985 by AG Hartenstein Kaiserslautern University

                        SET DEFAULT VALUES

 device     :

 directory  : [VLSI.PISA]

 filename   : REGARRAY

        started      at:    15:07:11.53          on:    19-MAR-1985
 device     :
 directory  : [VLSI.PISA]
 filename   : REGARRAY
```

figure 2.3 : PISA Set Default Values

```
                PISA SOFTWARE SYSTEM 4.1
  (C) Copyright 1985 by AG Hartenstein Kaiserslautern University

                   AUTOCHECK - DIALOG

   CIFfile is       : [VLSI.PISA]REGARRAY.CIF
   OUTPUTfiles are  : [VLSI.PISA]REGARRAY....

  Use default (Y)  : Y

        started      at:    15:07:11.53          on:    19-MAR-1985
  device     :
  directory  : [VLSI.PISA]
  filename   : REGARRAY
```

figure 2.4 : PISA Autocheck Dialog

```
                PISA SOFTWARE SYSTEM 4.1
  (C) Copyright 1985 by AG Hartenstein Kaiserslautern University

  Enter technology to be used :

    Available are :

    (1) NMOS MEAD & CONWAY
    (2) NMOS DORTMUND 1.5 MY
    (3) CMOS DORTMUND 1.5 MY

  Enter number of technology : 1

        started      at:    15:07:11.53          on:    19-MAR-1985
  device     :
  directory  : [VLSI.PISA]
  filename   : REGARRAY
```

figure 2.5 : PISA Technology Specification

```
PISA SOFTWARE SYSTEM 4.1
(C) Copyright 1985 by AG Hartenstein Kaiserslautern University

 Enter technology to be used :

   Available are :

   (1) NMOS MEAD & CONWAY
   (2) NMOS DORTMUND 1.5 MY
   (3) CMOS DORTMUND 1.5 MY

 Enter number of technology : 1
Job 129 entered on queue SYS$BATCH
```

```
      started      at:   15:07:11.53          on:   19-MAR-1985
 device    :
 directory : [VLSI.PISA]
 filename  : REGARRAY
```

figure 2.6 : PISA System Response

```
****************************************

            PISA normal end

(C) Copyright 1985 by AG Hartenstein Kaiserslautern University

     Date          : 19-MAR-1985
     Starttime     : 15:07:11.53
     Endtime       : 15:47:08.17
     used CPU-time : 000:00:04.7

****************************************
$
```

figure 2.7 : PISA Good Bye

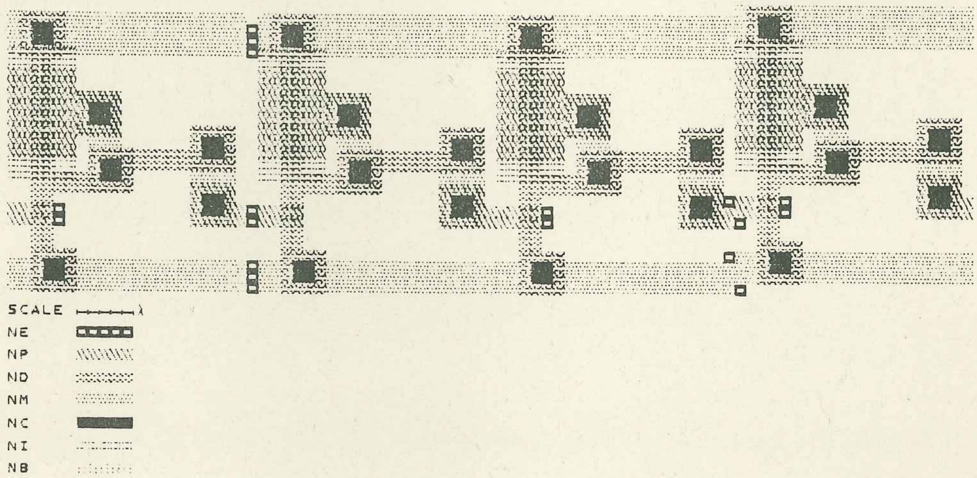


figure 2.8 : Stipple Plot

CHAPTER 3

FILES AND DATAFORMATS

The PISA SYSTEM uses the following file types :

- CIF Files (Manhattan CIF 2.0)
- FEM Files (Fully Expanded Manhattan CIF)
- Pixel Information Files (PIF)
- Pixel Attribute Files (PAF)
- Plot Files (plottercode to be plot on Hewlett Packard Plotter)
- Stip Files (stipple code to be print on DEC letterwriter LA100)
- DOMOS Files (input for the DOMOS circuit simulator)
- Technology Files

While PISA needs one of the above files a physical filename is required. Physical files are associated with logical files during runtime. PISA supports the user's filedialog.

Figure 3.1 shows the dataformat conversion within the PISA SYSTEM.

```
+-----> CIF
|        |
|        |
|        V
|       FEM ----------> PLOT
|        |
|        |
|        V
+---- PIF/PAF --------> STIP
         |
         |
         V
       DOMOS
```

figure 3.1 : dataformat conversion

3.1 CIF FILES

CIF (Caltech Intermediate Form) is a layout specification language. PISA supports only a subset of CIF 2.0, an extended version of Manhatten CIF 2.0.

Manhatten CIF is developed for description of orthogonal layout. Though the use of orthogonal wires and polygons simplifies mask layout design, these are processed. The following Berkeley User Extension are supported :

- 94 usertext

- 9 symbol name

A detailed syntax description is in appendix B.

3.2 FEM FILES

Contrary to the CIF format a FEM file contains a fully expanded layout description, i.e. subcell calls are decomposed. Wires and polygons are transformed to an equivalent sequence of boxes.

FEM is an intermediate format for plotter code - and pixel generation.

3.3 PIXEL INFORMATION FILES

PIF files contain the pixel-oriented layout description needed for the design rule check. The layout is mapped to a lambda grid, i.e. the layout consists of lambda squares. Every lambda square is represented by several bytes. The occurrence of a layer is marked by setting the corresponding bit.

3.4 PIXEL ATTRIBUTE FILES

PAF files contain a set of attributes necessary for PIF file processing. These are :

- layout size (window)
- used technology
- layout labels

3.5 PLOT FILES

These files contain plotter code for a HP plotter.

3.6 STIP FILES

STIP files contain stippelcode to drive a matrix printer with graphic extension. Currently code for the DEC letterwriter LA100 is generated.

3.7 DOMOS FILES

DOMOS files contain a netlist description of a layout. DOMOS is a simulator for electrical behavior of MOS circuits.

3.8 TECHNOLOGY FILES

TECHNOLOGY files contain informations about the used technology. These are :

- name of technology

FILES AND DATAFORMATS

- cif to lambda factor
- size of reference patterns
- used layers
- association of plotter pens with layers
- association of matrix patterns with layers
- design rules

CHAPTER 4

STRUCTURE OF THE PISA SYSTEM

Input for PISA is a CIF file, created with VAX/VMS editor or with the codeconverter KICtoCIF.

The program GENFEM converts a CIF file into an FEM file, which is used to generate plotter code or pixel files, respectively.

The program PLOTGEN generates plotter code. The program PIXGEN generates the pixel-oriented layout representation, needed for design rule checking.

The pixel format is input for the following programs :

- CHECK performs a design rule check for the specified technology. If CHECK detects design rule violations, errorfiles in FEM and CIF format are generated to localize errors in the layout.

- EXTRACT is a circuit extractor which generates input for MOS circuit simulator DOMOS.

- STIPGEN generates code for layout plots with matrix printers (currently DEC letterwriter LA100).

- CIFGEN generates CIF format for a given pixeloriented layout representation.

Figure 4.1 illustrates the data flow in PISA

```
                STRUCTURE OF THE PISA SYSTEM

            KIC resp. VAX/VMS editor
                      ^   ^
         +------------+   +------------+
         |                             V
   #############                 #############
+-># Error-CIF #                 # CIF-File  #<-------+
|  #############                 #############        |
|                                      |              |
|                                      |              |
|                                      V              |
|                                 +---------+         |
|                                 | GENFEM  |         |
|                                 +---------+         |
|                                      |              |
|                                      |              |
|                                      V              |
|  #############   +---------+   #############        |
+-># Error-FEM #-->| PLOTGEN |<--# FEM-File  #        |
|  #############   +---------+   #############        |
|                       |              |              |
|                       |              |              |
|                       V              V              |
|                ###############  +---------+         |
|                # PLOT-File   #  | PIXGEN  |         |
|                ###############  +---------+         |
|                       |              |              |
|                       V              |              |
|                    plotter           |              |
|                                      |              |
|                                      V              |
|                   +-------+     ###########    +---------+
+-------------------| CHECK |<----# PIF/PAF #--->| CIFGEN  |
                    +-------+     ###########    +---------+
                                     |   |
                        +------------+   +------------+
                        V                             V
                   +---------+                   +---------+
                   | STIPGEN |                   | EXTRACT |
                   +---------+                   +---------+
                        |                             |
                        |                             |
                        V                             V
                 ###############               ###############
                 # STIP-File   #               # DOMOS-File  #
                 ###############               ###############
                        |                             |
                        |                             |
                        V                             V
                  matrix printer                    DOMOS
```

figure 4.1 : PISA diagram

Figure 4.2 shows possible ways to create CIF files.

```
          +------------+
          |    KIC     |
          +------------+
                ^
                |
                V
          #############
          # KIC-File  #
          #############
              |   ^
         +----+   +------+
         V               |
  +------------+  +------------+    +------------------+
  | KICTOCIF   |  | CIFTOKIC   |    | VAX/VMS  Editor |
  +------------+  +------------+    +------------------+
         |               ^                   ^
         +-----------+   |  +----------------+
                     V   |  V
                  #############
                  # CIF-File  #
                  #############
```

figure 4.2 : Creation of CIF files

CHAPTER 5

WORKING WITH THE PISA SYSTEM

5.1 HOW TO GET STARTED

Start the PISA SYSTEM using the VMS command :

```
$ run disk$pisa:pisa
```

The terminal type has to be specified. At present two terminal types are supported :

- DEC VT100 and compatible terminals *)
  (screenoriented dialog)

- typewriter terminals
  (lineorientied dialog)

Figure 5.1 shows the hierarchical dialogue of the PISA SYSTEM entered after specification of the terminal type. The components of the dialogue are explained in the following paragraphs.

*) for example : Tektronix 4105 Color terminal

```
   +-------------------+
   | Top Command Level |
   +-------------------+
   +-------------------+
   | A(utocheck)       |
   +-------------------+
   | C(olorplot)       |
   +-------------------+
   | M(atrixplot)      |
   +-------------------+
+--+ EX(perimental)    |
|  +-------------------+
|
|  +-------------------+          +-------------------+
+--> Exp Command Level | +--------> CIF Command Level |
   +-------------------+ |        +-------------------+
   +-------------------+ |        +-------------------+
   | C(if)             +-+        | P(arse)           |
   +-------------------+          +-------------------+
   | F(em)             +-----+    | G(enfem)          |
   +-------------------+     |    +-------------------+
   | P(ix)             +---+ |
   +-------------------+   | |    +-------------------+
   | DR(c)             +-+ | +----> FEM Command Level |
   +-------------------+ | |      +-------------------+
                         | |      +-------------------+
                         | |      | PI(xgen)          |
                         | |      +-------------------+
                         | |      | PL(otgen)         |
                         | |      +-------------------+
                         | |
                         | |      +-------------------+
                         | +------> PIX Command Level |
                         |        +-------------------+
                         |        +-------------------+
                         |        | EX(tract)         |
                         |        +-------------------+
                         |        | C(ifgen)          |
                         |        +-------------------+
                         |
                         |        +-------------------+
                         +--------> DRC Command Level |
                                  +-------------------+
                                  +-------------------+
                                  | C(heck)           |
                                  +-------------------+
```

figure 5.1 : hierarchical dialog

5.2 TOP COMMAND LEVEL

The following commands are available :

A (utocheck)

Performs a batch-oriented processing of CIF sources, i.e. generation of FEM format, pixel format and design rule check. A plot file scaled for computer paper with errormarks is generated if the design rule checker detected design rule violations.

C (olourplot)

Performs a batch-oriented processing of a given CIF source, i.e. generation of FEM format and plotter code.

M (atrixplot)

Performs a batch oriented processing of CIF sources, i.e. generation of FEM format, pixel format and stippelcode to be printed on a DEC letterwriter LA100.

EX (perimental)

This command switches to the next lower dialog level. This level should be used only by experienced users.

ED (it)

Starts the VAX/VMS Edit/Edt editor.

DE (fault-set)

PISA requires the default value of a filename. It is useful to enter this command before any other command, because it simplifies working with PISA. We recommend urgently the specification of directory and filename.

SY (stem)

Interrupts the PISA SYSTEM for entering VAX/VMS commands.

H (elp)

Displays further information about the PISA SYSTEM.

EN (d)

Aborts the PISA SYSTEM.

5.2.1 EXPERIMENTAL COMMAND LEVEL

This dialogue level allows to start all functions of PISA interactively. The following commands are available :

ED (it)

Starts the VAX/VMS Edit/Edt editor.

C (if)

Switches to CIF Command Level.

F (em)

Switches to FEM Command Level.

P (ix)

Switches to PIX Command Level.

DR (c)

Switches to DRC Command Level.

DE (fault set)

see Top Command Level

SY (stem)

see Top Command Level

H (elp)

see Top Command Level

?

Displays further information about the current menue.

EN (d)

Leaves current menue, goes to TOP Command Level.

5.2.1.1 CIF COMMAND LEVEL

This command level provides CIF processing functions. These are :

P (arse)

Parses a given CIF source.

G (enfem)

Generation of FEM format for a given CIF source.

DE (fault set)

see TOP COMMAND LEVEL

SY (stem)

see TOP COMMAND LEVEL

H (elp)

see TOP COMMAND LEVEL

?

Displays further information about the current menue.

EN (d)

Leaves current menue, goes to Experimental Command Level.

5.2.1.2 FEM COMMAND LEVEL

This command level provides FEM processing functions. These are :

PI (xgen)

Generation of PIXEL format for a given FEM source.

PL (otgen)

Generation of PLOTTER code for a given FEM source.

DE (fault set)

see TOP COMMAND LEVEL

SY (stem)

see TOP COMMAND LEVEL

H (elp)

see TOP COMMAND LEVEL

?

Displays further information about the current menue.

EN (d)

Leaves the current menue, goes to Experimental Command Level.

5.2.1.3 PIX COMMAND LEVEL

This command level provides PIXEL processing functions (Pixel Information and Pixel Attribute Files). These are :

EX (tract)

Generates a netlist description in DOMOS format for given Pixel Information and Pixel Attribute files. The generated Domos file has to be completed by simulation parameters.

S (tipgen)

Generates stippel code for given Pixel Information and Pixel Attribute files.

C (ifgen)

Generates a CIF file for given Pixel Information and

Pixel Attribute files.

DE (fault set)

see TOP COMMAND LEVEL

SY (stem)

see TOP COMMAND LEVEL

H (elp)

see TOP COMMAND LEVEL

?

Displays further information about the current menue.

EN (d)

Leaves the current level, goes to Experimental Command Level.

5.2.1.4 DRC COMMAND LEVEL

This level provides the design rule checking function.

C (heck)

Starts the design rule checker.

DE (fault set)

see TOP COMMAND LEVEL

SY (stem)

see TOP COMMAND LEVEL

H (elp)

see TOP COMMAND LEVEL

?

Displays further information about the current menue.

EN (d)

Leaves the current menue, goes to Experimental Command Level.

CHAPTER 6

EXTRACTOR USER MANUAL

6.1 GENERAL DESCRIPTION

The program EXTRACT generates a net list description (DOMOS format) for a given pixel-oriented layout specification. A correct extractor output is only guaranteed for layouts, free of any design rule violations. So run the DRC prior using the extractor.
The recognition of butting contacts is not supported by the PISA System. We emphasize that the use of butting contacts will result in the generation of nonexistant transistors.
The user has to complete the output file, i.e. voltage sources, output lines and non rectangular transistor geometries.

<u>REMARK:</u>

DOMOS has been developed by H. Sibbert at Dortmund University in 1979.

6.2 INPUT RULES

This EXTRACTOR program needs labels to work correct. Labels have to be specified in the layout. Creating the labels, the user has to pay attention to the following rules:

1. All VDD nodes have to be labeled "VDD" literal (additional blanks).

2. All ground nodes have to be labeled "GND" literal (additional blanks).

3. All input nodes have to be labeled with the following format:

   "IN<nr>"

   <nr> means an integer between 1 and 99. Behind the number it isn't allowed to set any character (except blanks).

   ATTENTION : Don't use a number twice !!!

4. All output nodes have to be labeled with the following format:

   "OUT<nr>"

   <nr> means an integer between 1 and 99. Behind the number it isn't allowed to set any character (except blanks).

   ATTENTION : Don't use a number twice !!!

5. Label assignments have to be unique, i.e. a label must only be assigned to a point, where just one layer is present (exception of layer IMPLANT).

If one of these rules hasn't be considered, the program will detect a fatal error.

6.3 FILE TRANSFORMATION

After the program has ended successfully,

DON'T START THE DOMOS SIMULATION YET !!!

First, the created pseudo DOMOS file must be transformed.

1. On lines, including non rectangular transistors, the question marks must be substituted by the channel width and length of these transistors. To find these transistors in the layout, a command line follows which contains the transistor coordinates :

   $? <x> <y>

   <x> : absolute x-coordinate in the layout
   <y> : absolute y-coordinate in the layout

2. All voltage sources must be included into the file. To find them in the layout, all input nodes are marked in command lines :

   $= E<nr> <label>

   <nr> : number of the voltage source
   <label> : assigned layout label

3. To create the output line, it is necessary to know the output nodes. These nodes are noted in following command lines :

   $= N<nr> <label>

   <nr> : number of output variable
   <label> : assigned layout label

4.

   DON'T FORGET THE TIMER-, EXECUTE- AND END LINE !!!

6.4 IMPLEMENTING NOTES

At present, this program is implemented for a pixel-oriented file created by NMOS design rules. Don't try to use it with an other technology, it doesn't work !

APPENDIX A

LITERATURE

[1] P. Braun, E. Ewald, R. Hartenstein, J. Hassdenteufel, R. Hauck, A. Hirschbiel, M. Weber:
PISA - Pixel-oriented System for Layout Analysis of Kaiserslautern University;
Report, Fb. Informatik, Kaiserslautern University, 1983

[2] C. Mead, L. Conway:
Introduction to VLSI Systems;
Addison-Wesley, 1980

[3] P. Braun, R. Hartenstein, J. Hassdenteufel:
Pixel-oriented Layout Analysis: A semi-automatic Analyzer Generator for Design-Rule-Check and Circuit Extraction;
Fb. Informatik, Kaiserslautern University, 1983

[4] E. Ewald:
Entwicklung eines Mead-&-Conway Referenzmustersatzes;
Studienarbeit, Kaiserslautern University, 1984

[5] H. Landmann:
EQNTOTT: Equation to Truth-Table Format

[6] K. Jensen, N. Wirth:
PASCAL - User Manual and Report;
Springer, 1978

[7] W. Nebel:
CAD-Hilfsmittel zur Entwurfskontrolle hochintegrierter Digitalschaltungen;
Kaiserslautern University, 1984

[8] R. Hartenstein, R. Hauck, A. Hirschbiel,
W. Nebel, M. Weber :
PISA - a CAD package and special hardware for
pixel-oriented layout analysis;
Report; Kaiserslautern University, 1984

[9] A. Hirschbiel :
The PISA - machine ;
Diplomarbeit, Kaiserslautern University,
not yet published

[10] R. Hauck :
Ein Algorithmus zur Optimierung des
Raster-Scan Design Rule Checks ;
FB Informatik, Kaiserslautern University, 1984

[11] J. Bloedel :
PASCAL/VMS-Implementierung eines Programms zur
Erstellung von Layout-plots auf Matrixdruckern ;
FB Informatik, Kaiserslautern University, 1985

[12] M. Ryba :
Einfuehrung von Technologie-Files, Erweiterung des
Pixel-Formats ;
FB Informatik, Kaiserslautern University, 1985

[13] W. Nebel :
CAD-Entwurfskontrolle in der Mikroelektronik ;
BG Teubner, 1985

[14] R. Hartenstein, R. Hauck, A. Hirschbiel,
W. Nebel, M. Weber :
PISA - a CAD package and special hardware for
pixel-oriented layout analysis ;
Proceedings ICCAD, Santa Clara, CA, USA, 1984

[15] M. Weber :
Ein Patterngenerator zur automatischen
Referenzmustererzeugung ;
FB Informatik, Kaiserslautern University, 1984

[16] H. Salzmann :
Implementierung eines rasterorientierten Schaltkreisextraktors ;
FB Informatik, Kaiserslautern University, 1985

[17] H. Salzmann :
PISA : Extractor - User Manual ;
FB Informatik, Kaiserslautern University, 1985

[18] R. Hartenstein :
Das PISA - Programmsystem als schneller Design-Rule-Checker ;
FB Informatik, Kaiserslautern University, 1985

[19] R. Hartenstein :
EIS Verbundprojekt : Aufbruch in die Neue Mikroelektronik ;
Computer-Magazin, Juni 1984

APPENDIX B

MANHATTEN CIF SYNTAX

```
ManhattenCifFile        =   { { blank } [ command ] semi }
                              endCommand { blank } .

command                 =   primCommand |
                            defDeleteCommand |
                            defStartCommand semi
                              {{ blank } [ primCommand ] semi }
                              defFinishCommand .

primCommand             =   polygonCommand |
                            boxCommand |
                            wireCommand |
                            layerCommand |
                            callCommand |
                            userExtensionCommand |
                            commentCommand .

polygonCommand          =   "P" path .

boxCommand              =   "B" integer sep integer sep point
                              [ sep point ] .

wireCommand             =   "W" integer sep path .

layerCommand            =   "L" { blank } shortname .

defStartCommand         =   "D" { blank } "S" integer
                              [ sep integer sep integer ] .

defFinishCommand        =   "D" { blank } "F" .

defDeleteCommand        =   "D" { blank } "D" integer .

callCommand             =   "C" integer transformation .

userExtensionCommand    =   digit userText .

commentCommand          =   "(" commentText ")" .

endCommand              =   "E" .
```

MANHATTEN CIF SYNTAX

```
transformation        =    { { blank }
                              ( "T" point |
                                "M" { blank } "X" |
                                "M" { blank } "Y" |
                                "R" point
                              )
                            } .

path                  =    point { sep point } .

point                 =    sInteger sep sInteger .

sInteger              =    { sep } [ "-" ] integerD .

integer               =    { sep } integerD .

integerD              =    digit { digit } .

shortname             =    c [ c ] [ c ] [ c ] .

c                     =    digit | upperChar .

userText              =    { userChar }

commentText           =    { commentChar } |
                           commentText "(" commentText ")"
                             commentText.

semi                  =    { blank } ";" { blank }.

sep                   =    upperChar |
                           blank.

digit                 =    "0" | "1" | "2" | "3" | "4" |
                           "5" | "6" | "7" | "8" | "9" .

upperChar             =    "A" | ... | "Z" .

blank                 =    any ASCII character except digit,
                             upperChar, "-", "(", ")", or ";" .

userChar              =    any ASCII character except ";" .

commentChar           =    any ASCII character except "("
                             or ")" .
```